

REMARKS

This is intended as a full and complete response to the Final Office Action dated January 13, 2009, having a shortened statutory period for response set to expire on April 13, 2009. Applicants submit this response to place the application in condition for allowance or in better form for appeal. Please reconsider the claims pending in the application for reasons discussed below.

Claims 1-3, 5-6, 9-12, 14-18, 20-21 and 24-27 are pending in the application. Claims 1-3, 5-6, 9-12, 14-18, 20-21 and 24-27 remain pending following entry of this response.

Claim Rejections - 35 U.S.C. § 103

Claims 1-3, 5, 9, 12, 14, 16-18, 20 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant Admitted Prior Art in Applicant's Specification (paragraphs 8, 9 and 35) ("AAPA") in view of *Veditz, et al.* ("*Veditz*") U.S. Patent No. 6,496,793 and further in view of *Horn, et al.* ("*Horn*"), U.S. Publication No. 2002/0156688. Applicants respectfully traverse this rejection.

In this case, Applicant submits that the cited references do not disclose "a method of determining an appropriate character set for use in client-server communications" that includes:

- (a) selecting a character set for a client request made by client to a server using a network communication protocol, the selecting comprising:
 - determining whether the client request includes, as part of the network communication protocol, a request character set designation; and
 - if the client request does not include the request character set designation:
 - (i) retrieving locale information contained in the client request;
 - (ii) selecting a character set to assign to the request character set designation by associating the locale information with the request character set designation using mapping data located on the server; and
 - (iii) associating the request character set designation with a first code-set converter designation, wherein the first code-set converter designation is contained in a lookup table and is mapped

in the lookup table with the character set assigned to the request character set designation, and wherein a first code-set converter corresponding to the first code-set converter designation maps characters of the request character set designation to corresponding characters of the first code-set converter designation while processing the request.

Claims 12 and 16 recite a similar limitation. Regarding the last quoted limitation, the Examiner concedes that “AAPA” does not disclose this limitation, but turns to *Veditz*. In particular, the Examiner suggests:

See [Veditz] fig. 28 and 2C - i.e. “LDID Value”; see also col. 13, lines 1-67 to col. 14, lines 1-62 where each character set is associated with a code-set designation in a lookup table that maps the associations.

Final Office Action, p. 4. However, columns 13 and 14 of *Veditz* describe a header file of definitions specifying a relationship between what *Veditz* describes as “language driver” and a “code page.” In other words, the mappings specify “for language driver X use code page Y”. As used in *Veditz*:

“Language drivers” are provided to correctly handle characteristics of a given language. The drivers reference a character set and a collection of tables describing the rules for that character set. For instance, language drivers include information about character sets (code pages), sorting orders, upper case and lower case rules, which characters are alphabetic, and what double-letter combination it is to accept.

Veditz, 11:59-65. And *Veditz* describes a “Language Driver Identifier” or “LDID” as an element used to identify a given “Language Driver;” specifically:

Particularly for those embodiments having data objects constrained by downward compatibility or storage space considerations, the descriptor is a Language Driver Identifier (LDID) of the present invention. The LDID may be embodied in the form of a system-comparable unit, such as an ID byte which references an agreed-upon set of values (e.g., locale lookup table). For purposes of clarity, the discussion which follows will focus on use of the LDID descriptor embodied as a byte identifier.

Veditz, 12:43-53. Thus, the “LDID” value cited to be the Examiner – merely provides a byte value used to identify a particular “Language Driver.” *Veditz*, col. 13 provides a list of language driver identifiers which provide a “one-to-one correspondence between a language driver and its LDID.” At the same time, this table does not disclose anything about mapping from **a first character set** to **a second character set** in general, and

does not disclose the claimed limitation of “a first code-set converter corresponding to the first code-set converter designation” which “maps characters of the request character set designation to corresponding characters of the first code-set converter designation while processing the request.” Instead, the LDID value simply identifies a “language Driver,” which *Veditz* explicitly defines as a “reference a character set and a collection of tables describing the rules for that character set. For instance, language drivers include information about character sets (code pages), sorting orders, upper case and lower case rules, which characters are alphabetic, and what double-letter combination it is to accept.” *Veditz*, 11:59-65. Nothing in this generic description of “rules for a character set” does *Veditz* disclose the narrow, focused limitation of:

associating the request character set designation with a first code-set converter designation, wherein the first code-set converter designation is contained in a lookup table and is mapped in the lookup table with the character set assigned to the request character set designation, and wherein a first code-set converter corresponding to the first code-set converter designation maps characters of the request character set designation to corresponding characters of the first code-set converter designation while processing the request.

Accordingly, for all the foregoing reasons, Applicants submit that Applicant “admitted prior art” in view of *Veditz*, does not disclose the limitations of claims 1, 12, or 16. Therefore, these claims (and claims dependent therefrom) are believed to be allowable, and allowance of the claims is respectfully requested.

Claims 6, 10, 11, 21, 26 and 27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant Admitted Prior Art in Applicant’s Specification (paragraphs 8, 9 and 35) (“AAPA”) in view of *Veditz, et al. (“Veditz”)* U.S. Patent No. 6,496,793.

Claims 15 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant Admitted Prior Art in Applicants’ Specification (paragraphs 8, 9 and 35) (“AAPA”) in view of *Veditz, et al. (“Veditz”)* U.S. Patent No. 6,496,793 and further in view of *Kan, et al. (“Kan”)*, U.S. Publication No. 2003/0088544).

Claims 6, 10, 11, 21, 26, and 27 each depend from one of independent claims 1 or 16. As the discussion above demonstrates that Applicant “admitted prior art” in view

of *Veditz*, does not disclose the limitations of these independent claims, Applicants submit that dependent claims 6, 10, 11, 21, 26, and 27 are allowable without the need for further comment.

Claims 6, 10, 11, 21, 26, and 27 each depend from one of independent claims 1 or 16. As the discussion above demonstrates that Applicants “admitted prior art” in view of *Veditz*, does not disclose the limitations of these independent claims, Applicants submit that dependent claims 6, 10, 11, 21, 26, and 27 are allowable without the need for further comment.

Claims 1, 3, 5, 9, 12, 14, 16, 18, 20 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz, et al.* (“*Veditz*”) U.S. Patent No. 6,496,793, in view of *Watanabe, et al.* (“*Watanabe*”), U.S. Patent No. 6,185,729. Respectfully, Applicants traverse this rejection.

In particular, Applicants submit that the combination of *Veditz* and *Watanabe* do not teach, show, or suggest a “method of determining an appropriate character set for use in client-server communications” that includes “determining whether the client request includes, as part of the network communication protocol, a request character set designation, and if the client request does not include the request character set designation: (iii) associating the request character set designation with a first code-set converter designation, wherein the first code-set converter designation is contained in a lookup table and is mapped in the lookup table with the character set assigned to the request character set designation, and wherein a first code-set converter corresponding to the first code-set converter designation maps characters of the request character set designation to corresponding characters of the first code-set converter designation while processing the request”.

Like the rejection based on “Applicants Admitted Prior Art” and *Veditz*, the Examiner suggests:

[*Veditz* discloses] (iii) associating the character set with a code-set converter designation, (See fig. 28 and 2C - i.e. “LDID Value”; see also col. 13, lines 1-67 to col. 14, lines 1-62 where each character set is

associated with a code-set designation in a lookup table that maps the associations).

Final Office Action, p. 10. For all the reasons set forth above regarding these same passages, the LDID value provides an identifier for a “language driver;” nothing about the table in *Veditz*, col. 12-13 which lists a “one-to-one correspondence between a language driver and its LDID,” discloses anything about mapping from **a first character set** to a **second character set** in general, and does not disclose the claimed limitation of “a first code-set converter corresponding to the first code-set converter designation” which “maps characters of the request character set designation to corresponding characters of the first code-set converter designation while processing the request.” Instead, the LDID value may be used to indentify a “language Driver, and as stated the language driver “reference a character set and a collection of tables describing the rules for that character set.”

Further, *Veditz* does not teach the claimed step of “determining whether a client request includes ... a request character set designation, and if the client request does not include the request character set designation, (i) retrieving locale information contained in the client request and (ii), selecting a character set to assign to the request character set designation by associating the locale information with the request character set designation using mapping data located on the server.

The Examiner suggests:

[*Veditz* discloses] selecting a character set to assign to the request character set designation by associating the locale information with the request character set designation using mapping data located on the server (Fig. 2B - if Active LDID is not equal to Local LDID it maps the Local LDID into the Active LDID; see also col. 3, lines 54-60; col. 7, lines 5264; col. 18, lines 21-26); and

Office Action, p. 11. But the mappings described in Figure 2B suggest that an “active” LDID (i.e., the LDID being used to process an object) may be updated to match the “local” LDID). That is, when a system is currently using a particular language driver (the active LDID) and receives an object with a different default language driver (the local LDID), the system changes the active LDID to match.

More generally, the system of Veditz maintains language configuration to determine when the system is inappropriately configured for a data object about to be processed. *Veditz*, 7:45-50. Each data object in the system may be tagged with an LDID, and the system itself maintains an “active” LDID (i.e., the LDID presently being used by the system). *Veditz*, 14:56-62. The active LDID, in turn, is written to data objects which the system “touches” (i.e., creates or modifies). *Veditz*, 14:63-64. In the event of a mismatch between the “active” LDID and the LDID of a “data object,” corrective actions may be taken. Further, the system of *Veditz* requires data objects **to include** an LDID value in order to function. The present claims, however, are directed to determining a character set when a “client request” **explicitly lacks** a character set designation. Thus, Applicants submit that Veditz does not disclose a method for processing a request that “determining whether the client request includes, as part of the network communication protocol, a request character set designation; **and if the client request does not include the request character set designation [performing subsequent steps].**”

Further, claims 1 and 16 recite: “retrieving locale information contained in the client request” In other words, when a character set designation is not included in the request, retrieve something else *from the client request*; in this case “locale information” contained in the client request. On this point, the Examiner cites *Veditz* Figure 3B and states “Fig. 3B → compares LDID of data file to Active LDID; see also col. 3, lines 29-31.” See *Final Office Action*, p. 12. The passage cited by the Examiner, however, merely describes that the LDID may be represented using a byte (i.e., 8 bits of data):

The LDID, which may be in the form of an ID byte, references a set of language driver values (e.g., lookup table of locales).

Figure 3 is a flow chart “illustrating a language-dependent file operation method of the present invention.” *Veditz*, 4:20-21. The method generally includes comparing a data value from a database header file with the current “active LDID” maintained by the system. If no “LDID value” is included in the data object, *Veditz* discloses that a

database object may be opened using a read-only access mode. See, e.g., *Veditz*, Figure 3A, 304.

If an “LDID value” is stored with the data object, *Veditz* discloses comparing this value with the “active LDID”. See e.g., *Veditz*, Figure 3B. The “Active LDID” value, however, is not “retrieved from the client request” as recited by claims 1, 12, and 16; instead the “active LDID” is a value maintained by the system for “tagging” each “data object” modified during a given session. For example, *Veditz*, Figure 2A illustrates the “active LDID” element as part of the language configurator 230. Separately, Figure 2B also illustrates a data object 201, with a local LDID 215 in a header file. Even assuming that the “data object” illustrated in Figure 2 reads on the recited “client request,” clearly the “active LDID” is not retrieved **from the data object 201**. Quite the contrary, the “active LDID” is a system variable independent of any particular data object or request. Thus, Applicants respectfully submit that the material cited from *Veditz* does not teach, show, or suggest that limitations recited by claims 1, 12, and 16.

Claims 2, 6, 10, 11, 17, 21, 26 and 27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz, et al.* (“*Veditz*”) U.S. Patent No. 6,496,793, in view of *Watanabe, et al.* (“*Watanabe*”), U.S. Patent No. 6,185,72, and further in view of *Horn, et al.* (“*Horn*”), U.S. Publication No. 2002/0156688.

Claims 15 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Veditz, et al.* (“*Veditz*”) U.S. Patent No. 6,496,793, in view of *Watanabe, et al.* (“*Watanabe*”), U.S. Patent No. 6,185,72, and further in view of *Kan, et al.* (“*Kan*”), U.S. Publication No. 2003/0088544.

Claims 15 and 25 each depend from one of independent claims 12 or 16. As the discussion above demonstrates that *Veditz*, in view of *Watanabe*, does not disclose the limitations of these independent claims, Applicants submit that dependent claims 15 and 25 are allowable without the need for further comment.

Conclusion

Having addressed all issues set out in the office action, Applicants respectfully submit that the claims are in condition for allowance and respectfully request that the claims be allowed.

If the Examiner believes any issues remain that prevent this application from going to issue, the Examiner is strongly encouraged to contact Gero McClellan, attorney of record, at (336) 698-4286, to discuss strategies for moving prosecution forward toward allowance.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. MCCLELLAN, Reg. #44227/

Gero G. McClellan
Registration No. 44,227
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicants